Single Machine Scheduling with Job Rejection and Generalized the Number of Rejected Jobs

Ruiqing Sun*, Mingyao Li, Bin Deng

School of Mathematics and Statistics, Yunnan University, Kunming, China.

* Corresponding author. Email: ruiqingsun@mail.ynu.edu.cn Manuscript submitted January 14, 2022; accepted March 3, 2022. doi: 10.17706/ijapm.2022.12.3.22-33

Abstract: In this paper, we consider the single machine scheduling with job rejection and generalized the number of rejected jobs. More specifically, each job is either accepted and processed on the single machine, or is rejected by paying a rejection cost, but the total number of rejected jobs are strictly upper bounded by given thresholds. Two problems are considered: (1) minimize total weighted completion time and the sum of rejection cost, and (2) minimize total weighted completion time under the job rejection constraint. For the first scheduling problem, we provide a 2.618-approximation algorithm, a pseudo-polynomial time algorithm and a full polynomial-time approximation scheme (FPTAS) for this problem. In additions, we also discuss some special cases and provide two polynomial-time algorithms for them. For the latter problem, we provide a pseudo-polynomial time dynamic programming algorithm. Furthermore, we develop the dynamic programming into a fully polynomial time approximation scheme.

Key words: Scheduling, rejection, dynamic programming, fully polynomial time approximation scheme.

1. Introduction

In the classical deterministic scheduling problem, it is assumed that all jobs must be processed. However, in many practical cases, mainly in high load production to order systems, a manufacturer often receive a great deal of orders (jobs) from the customers. Due to the lack of enough resources such as machines and operators, accepting all jobs may lead to delayed order completion, which in turn may lead to high inventory and tardiness costs. Thus, sometimes the decision-maker may only accept some jobs and rejects the others, scheduling the accepted jobs to the machines for processing. More specifically, for each job decision-makers must decide either to schedule the job or to reject it. When a job is rejected, a corresponding rejection cost is required. Thus, from the practical point of view, rejecting some jobs can save time and reduce costs. We say that this kind of problem is job rejection scheduling.

Scheduling with job rejection was first proposed by Bartal *et al.* [1], they considered parallel-machine scheduling problems with rejection, where the objective is to minimize the makespan of the accepted jobs plus the total penalty of the rejected jobs and they presented a PTAS and FPTAS when the number of machines is arbitrary and fixed, respectively. Then, Ou *et al.* [2] studied the same model in [1] and presented a $(3/2 + \varepsilon)$ -approximation algorithm with running time $O(n \log n + n/\varepsilon)$, where ε is a small given positive constant. This result was further improved by Ou and Zhong [3] who designed a $(4/3 + \varepsilon)$ -approximation algorithm with running time $O(n \log n + n/\varepsilon)$, where scheduling problems with the objective of minimizing the maximum tardiness/lateness of the accepted jobs plus the total penalty of

the rejected jobs. Zhang *et al.* [5] studied a scheduling model with release dates to minimize the makespan of the accepted jobs plus the total penalty of the rejected jobs, they proved that this problem is NP-hard and presented a fully polynomial time approximation scheme. He et al. [6] and Ou et al. [7] independently designed an improved approximation algorithm with a running time of $O(n \log n)$. At the same time, Zhang et al. [8] also considered the parallel-machine case of the above problem: given a dynamic programming algorithm with pseudo-polynomial. When the number of machines m is not fixed, they presented a 2-approximate algorithm. When the number of machines *m* is a fixed constant, they presented a FPTAS. In recent years, many people consider new problems with rejection. For example, the vector scheduling problem, where each job J_j is associated with a *d*-dimensional vector $p_j = (p_j^1, p_j^2, \dots, p_j^d)$. The problem aims to schedule n d -dimensional jobs on m machines and the objective is to minimize the maximum load over all dimensions and all machines. Li and Cui [9] considered a single machine vector scheduling problem with rejection that aims to minimize the maximum load over all dimensions plus the sum of the penalties of the rejected jobs. They proved that this problem is NP-hard and designed a combinatorial *d* -approximation algorithm and $\frac{e}{e-1}$ -approximation algorithm based on randomized rounding. Next, Dai and Li [10] studied vector scheduling problem with rejection on two machines and designed a combinatorial 3-approximation and 2.54-approximation algorithm based on randomized rounding. Then, Liu et al. [11] studied single machine vector scheduling with general penalties and propose a noncombinatorial $\frac{e}{e-1}$ -approximation algorithm and a combinatorial min $\{r, d\}$ -approximation algorithm, where r is the maximum ratio of the maximum load to the minimum load on the d -dimensional vector. More new scheduling with rejection, for example, with submodular penalties can be found in ([12]-[14]).

For the scheduling problems under the job rejection constraint model. Cao et al. [15] considered the single-machine scheduling problem under the job rejection constraint to minimize the total weighted completion times of the accepted jobs. They showed that this problem is binary NP-hard and presented a pseudo-polynomial-time dynamic programming algorithm and a fully polynomial-time approximation scheme. Zhang et al. [16] extended the study of [15] to multiple identical parallel machines and presented a FPTAS. Zhang et al. [17] considered the parallel-machine scheduling problem under the job rejection constraint to minimize the makespan of the accepted jobs. They showed that this problem is NP-hard and presented a pseudo-polynomial-time dynamic programming algorithm and two fully polynomial-time approximation schemes. This result was further improved by Li et al. [14] who designed a FPTAS with running time $O(1/\varepsilon^{2m+3} + mn^2)$. Zhang *et al.* [17] also considered the single-machine scheduling problem with release dates under the job rejection constraint to minimize the makespan and presented a fully polynomial time approximation scheme. Liu et al. [18] consider the single machine parallel-batch scheduling problem with release dates and the number of rejected jobs not exceeding a given threshold. The objective function is to minimize the sum of the makespan and the total rejection cost. They propose a pseudo-polynomial time dynamic programming exact algorithm, a 2-approximation algorithm and a fully polynomial time approximation scheme. More papers dealing scheduling under the job rejection constraint are ([19], [20]).

For the objective is to minimize the total weighted completion time and the total rejection cost. Engels *et al.* [21] studied the problem of minimizing the total weighted completion time of the accepted jobs and the total rejection penalty of the rejected jobs, they proved that the problem is NP-hard and presented a FPTAS for the problem. They reduced the problem $1|rej|\sum w_jC_j + \sum e_j$ and $1|rej,r_j| \sum w_jC_j + \sum e_j$ to the scheduling problem $R|rej|\sum w_jC_j + \sum e_j$ and $R|rej,r_j| \sum w_jC_j + \sum e_j$. Using the existing results of

unrelated machine scheduling problem, they obtained that there are 3/2-approximation algorithms and 2-approximation algorithms for problems $1|rej|\sum w_jC_j + \sum e_j$ and $1|rej,r_j|\sum w_jC_j + \sum e_j$, respectively. For more results on scheduling with rejection, we refer the readers to the surveys provided by Shatbay *et al.* [22] and Zhang [23], respectively.

Motivated by the studies in ([15], [18], [21]). In this paper, we study the single machine scheduling with job rejection under the number of rejected jobs constraint and present some interesting results. The remaining parts are organised as follows: In Section 2, we provide the problem formulation on our problems. In Section 3, we first provide a 2.618-approximation algorithm and a pseudo-polynomial time algorithm, furthermore, we develop the pseudo-polynomial time algorithm into a full polynomial-time approximation scheme (FPTAS) for our first problem. In addition, we also discuss some special cases of our first problem and provide two polynomial-time algorithms for them. In Section 4, we provide a pseudo-polynomial time algorithm. Furthermore, we develop the pseudo-polynomial time algorithm into a full polynomial-time approximation scheme (FPTAS) for our first problem. In addition, we also discuss some special cases of our first problem and provide two polynomial-time algorithms for them. In Section 4, we provide a pseudo-polynomial time algorithm into a full polynomial-time algorithm. Furthermore, we develop the pseudo-polynomial time algorithm into a full polynomial-time approximation scheme (FPTAS) for our second problem. We conclude the paper in Section 5, and suggest some possible future research.

2. Preliminaries

In this section, we formally define our rejection scheduling problems. Given a set of n jobs $J = \{1, 2, \dots, n\}$, with processing time p_j , weights w_j and rejection penalties e_j of the job, respectively, for $j = 1, 2, \dots, n$. For each job, we either process it or reject it, but the total number of rejected jobs cannot exceed a given threshold of K. If we schedule job j, we denote its completion time by C_j . If we reject the jobs j, we pay the rejection penalties e_j . Then we define the set of processed jobs as A, and the set of rejected jobs as , let |R| represents the number of jobs in the set of R. In order to ensure a certain service level, the total number of rejected jobs shall not exceed a given value K, where the K is a fixed constant. We introduce generalized the number of rejected jobs into our two problems as follows:

For the first problem, our goal is to minimize the sum of the total weighted completion time of processed jobs and the total penalty of rejected jobs. Our first problem is to generalize the problem proposed by [21], and if K = n, the problem proposed by [21]. In [21], problem $1 |rej| \sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$ is NP-hard, thus our problem is NP-hard, too. Bases on the three-field notation $\alpha |\beta| \gamma$ introduced by Graham *et al.* [24], this problem is denoted by:

$$1 | rej, | R | \leq K | \sum_{j \in A} w_j C_j + \sum_{j \in R} e_j.$$

$$\tag{1}$$

For the second problem, the objective is to find a schedule to minimize the total weighted completion time of processed jobs under the job rejection constraint $\sum_{j \in R} e_j \leq U$ where U is a given upper bound. Our second problem is to generalize the problem proposed by [15], and if K = n, the problem proposed by [15]. In [15], problem 1 | rej, $\sum_{j \in R} e_j \leq U | \sum_{j \in A} w_j C_j$ is NP-hard, thus this problem is NP-hard, too. Bases on the three-field notation $\alpha | \beta | \gamma$ introduced by Graham *et al.* [24], this problem is denoted by:

$$1 | rej, | R | \leq K, \sum_{j \in R} e_j \leq U | \sum_{j \in A} w_j C_j.$$

$$(2)$$

3. Generalized the Number of Rejected Jobs

In this section, we consider the problem $1|rej, |R| \le K| \sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$. We provide a 2.618-approximation algorithm. Inspired by [21], we also solve this problem base on the dynamic programming algorithm, and then modify it to obtain a fully polynomial time approximation scheme (FPTAS). In addition, we also discuss some special cases of our problem.

When job rejection is not allowed, the corresponding scheduling problem can be denoted by $1 \| \sum w_j C_j$. For problem $1 \| \sum w_j C_j$, Smith [25] showed that the WSPT rule yields an optimal schedule. The WSPT rule can be stated as follows:

WSPT Rule: Schedule the jobs in order of non-decreasing p_i/w_i .

Lemma 3.1. For problem $1 | rej, |R| \le K | \sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$, there is an optimal schedule such that all accepted jobs are processed in the WSPT rule.

3.1. A 2.618-Approximation Algorithm

In this subsection, we provide a 2.618-approximation algorithm for problem $1|rej, |R| \le K |\sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$. Sort all jobs in ascending order of p_j/w_j . Let $x_j = 1$ if job J_j is accepted and $x_j = 0$ if job J_j is rejected. Thus, problem $1|rej, |R| \le K |\sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$ is equivalent to the following Integer Linear Programming (ILP).

Min
$$\sum_{j=1}^{n} w_j C_j + \sum_{j=1}^{n} (1 - x_j) e_j$$

s.t. $C_j = x_j \sum_{k=1}^{j} x_k p_k, j = 1, \dots, n,$
 $\sum_{j=1}^{n} (1 - x_j) \le K,$
 $x_j \in \{0, 1\}, j = 1, \dots, n.$

For each rejected job J_j , we have $C_j = 0$. If we replace $x_j \in \{0,1\}$ by $0 \le x_j \le 1$ for each $j = 1, \dots, n$, we can obtain a Relaxed Linear Programming (RLP).

Min
$$\sum_{j=1}^{n} w_j C_j + \sum_{j=1}^{n} (1 - x_j) e_j$$

s.t. $C_j = x_j \sum_{k=1}^{j} x_k p_k, j = 1, \dots, n,$
 $\sum_{j=1}^{n} (1 - x_j) \le K,$
 $0 \le x_i \le 1, j = 1, \dots, n.$

Algorithm A_1

Step 1: Solve the RLP. Let (x_1^*, \dots, x_n^*) be an optimal solution of RLP. If $x_j^* \ge \frac{\sqrt{5}-1}{2}$, then we set $x_j = 1$; otherwise, we set $x_j = 0$.

Step 2: Accept all jobs with $x_j = 1$ and reject all jobs with $x_j = 0$. Process all accepted jobs in the WSPT rule.

Let π be the schedule obtained from algorithm A_1 . Furthermore, we let Z and Z^* be the corresponding

objective values of π and an optimal schedule π^* , respectively.

Theorem 3.2
$$Z \le \frac{3+\sqrt{5}}{2}Z^*$$
.

For each rejected job J_j , we have $C_j = 0$. And note that $w_j C_j = w_j x_j \sum_{k=1}^j x_k p_k \le \frac{3+\sqrt{5}}{2} w_j x_j^* \sum_{k=1}^j x_k^* p_k$, for each

job J_j . That is, $\sum_{j=1}^n w_j C_j \le \frac{3+\sqrt{5}}{2} \sum_{j=1}^n w_j C_j^*$. Furthermore, we also have $\sum_{j \in R} e_j = \sum_{j=1}^n (1-x_j) e_j \le \frac{3+\sqrt{5}}{2} \sum_{j=1}^n (1-x_j^*) e_j$.

Thus, we have $Z = \sum_{j=1}^{n} w_j C_j + \sum_{j \in R} e_j \le \frac{3 + \sqrt{5}}{2} \sum_{j=1}^{n} w_j C_j^* + \frac{3 + \sqrt{5}}{2} \sum_{j=1}^{n} (1 - x_j^*) e_j \le \frac{3 + \sqrt{5}}{2} Z^*$ This completes the proof

of Theorem 3.2.

3.2. Dynamic Programming

In this subsection, we give an $O(nK\sum_{j=1}^{n}w_j)$ time algorithm using dynamic programming to solve problem $1|rej, |R| \le K |\sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$.

Similar to [21], to solve our problem, we set up a dynamic program for a harder problem: namely, to find the schedule that minimizes the objective function when the total weight of the scheduled jobs and the total number of rejected jobs are given. We number the jobs in ascending order of p_j/w_j . Let $\phi_{k,W,j}$ denote the optimal value of the objective function when the jobs in consideration are $j, j+1, \dots, n$, the total number of reject jobs is k, and the total weight of the scheduled jobs is W.

Now, consider any optimal schedule for the job $j, j+1, \dots, n$, in which the total number of reject jobs is k, the total weight of the scheduled jobs is W. In any such schedule, there are two possible cases, either job j is rejected or job j is scheduled.

Case 1. Job *j* is rejected. Then, the optimal value of the objective function is clearly $\phi_{k-1,W,j+1} + e_j$, since the total weight of the scheduled jobs among $j+1, \dots, n$ must be *W*, the total number of reject jobs is k-1.

Case 2. Job *j* is scheduled. In this case, the total weight of the scheduled jobs among $j+1, \dots, n$ must be $W - w_j$. Also, when job *j* is scheduled before all jobs in the optimal schedule for jobs $j+1, \dots, n$, the completion time of every scheduled job among $j+1, \dots, n$ is increased by p_j . Then, the optimal value of the objective function is $\phi_{k,W-w_i,j+1} + Wp_j$.

Combining the above two cases, we have following dynamic programming algorithm DP1:

Dynamic Programming Algorithm DP1

The Boundary Conditions:

$$\begin{cases} \phi_{k,W,n} = +\infty, k = 0, W \neq w_n; \\ \phi_{k,W,n} = e_n, k \neq 0, W \neq w_n; \\ \phi_{k,w_n,n} = w_n p_n, k = 0; \\ \phi_{k,w_n,n} = \min\{w_n p_n, e_n\}, k \neq 0. \end{cases}$$

The Recursive Function:

$$\phi_{k,W,j} = min(\phi_{k-1,W,j+1} + e_j, \phi_{k,W-w_j,j+1} + Wp_j).$$

Since the reject of the jobs can be at most *K*, the weight of the scheduled jobs can be at most $\sum_{j=1}^{n} w_j$,

and the answer to our original problem is $min\{\phi_{k,W,1}: 0 \le k \le K, 0 \le W \le \sum_{j=1}^{n} w_j \}$.

Theorem 3.3. Dynamic programming yields an $O(nK\sum_{j=1}^{n}w_j)$ time algorithm for solving problem 1 | rej, $|R| \le K |\sum_{i \in A} w_j C_j + \sum_{i \in R} e_j$.

3.3. Fully Polynomial Time Approximation Scheme

In this subsection, we describe a fully polynomial time approximation scheme (FPTAS) for problem $1 | rej, | R | \le K | \sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$. There is a number of works concerning designing FPTASes in general, e.g. Woeginger [26]. A survey of Polynomial Approximation Schemes minimizing the total completion time of a schedule can be found in Afrati and Milis [27].

Similar to [21], we identify any given schedule its ε' -aligned schedule by sliding the scheduled time of each job (starting from the first scheduled job and proceeding in order) forward in time until its completion time coincides with the next time instant of the form $\tau_i = (1 + \varepsilon')^i$, the job is then said to be ε' -aligned. Note that when we slide the scheduled time of job i, the scheduled times of later jobs also get shifted forward in time by the same amount as for job i. When the time comes to ε' -aligned job j, its completion time has already moved forward in time by an amount equal to the sum of the amounts moved by the completion times of the jobs scheduled earlier than itself. ε' -aligned schedule may contain idle time. Without any loss of generality, we can assume that the smallest processing time is at least 1. Otherwise, we can divide each processing time p_j and rejection penalty e_j by the smallest processing time p < 1.

Lemma 3.4. For problem $1 | rej, | R | \le K | \sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$, the optimal objective function value increases by a factor of at most $(1 + \varepsilon')^n$ for any $\varepsilon' > 0$, when we restrict our attention to ε' -aligned schedules only.

Proof: The prove follows same idea as in [21]. We prove the conclusion by induction on *i*. Note that if a job finishes at time $t \in (\tau_i - 1, \tau_i]$ after all the jobs before it have been ε' -aligned, its completion time after being ε' -aligned is $\tau_i < (1+\varepsilon')t$. Since we assume that the minimum processing time is at least 1, and the smallest τ_i is $\tau_0 = 1$, clearly C_1 increases by a factor of $(1+\varepsilon')$. Now, assume by induction, that C_i increases by a factor of at most $(1+\varepsilon')^i$. When the turn comes to ε' -align the *i*+1 th scheduled job, its completion time C_{i+1} has already increased by an amount equal to the amount moved by C_i , which is at most $[(1+\varepsilon')^i -1]C_i$. When it is ε' -aligned, C_{i+1} further increases by a factor of at most $(1+\varepsilon')$. Thus, the final value of C_{i+1} is at most $(1+\varepsilon')^i -1]C_i + C_{i+1}$. Since $C_i < C_{i+1}$, this is at most $(1+\varepsilon')^{i+1}C_{i+1}$. Since the number of scheduled jobs is at most n, the result follows.

Let $\varepsilon' = \varepsilon/(2n)$, the optimal objective function value increases by a factor of at most $(1+\varepsilon)$ for any $\varepsilon > 0$, since $(1+\varepsilon/2n)^n \le (1+\varepsilon)$. For our FPTAS, we set up a dynamic program for a harder problem: namely, to find the ε' -aligned schedule that minimizes the objective function when the total number of reject jobs less than or equal to k and the completion time of the latest scheduled job is on or before τ_i , for a given i. We number the jobs in ascending order of p_j/w_j . Let $\phi_{k,i,j}$ denote the optimal value of the objective function when the jobs in consideration are $1, 2, \dots, j$, the total number of reject jobs less than or equal to k and the latest scheduled job (if any) in an ε' -aligned schedule completes at time less than or equal to τ_i ($i \ge 0$).

Now, consider any optimal schedule for the jobs $1, 2, \dots, j, j+1$, in which the total number of reject jobs

less than or equal to k and the latest scheduled job completes by time τ_i . In any such schedule, there are two possible cases, either job j+1 is rejected or job j+1 is scheduled.

Case 1. Job j+1 is rejected. Then, the optimal value of the objective function is $\phi_{k-1,i,j} + e_{j+1}$, since the last of the scheduled jobs among 1, 2, ..., *j* must finish at time less than or equal to τ_i , the total number of reject jobs less than or equal to k-1.

Case 2. Job j+1 is scheduled. This is possible only if $p_{j+1} \le \tau_i$. In this case, if there was a job scheduled before job j+1, it must have completed at time $\tau_{i'}$, where $(\tau_i - \tau_{i'}) \ge p_{j+1}$. Then, the optimal value of the objective function is $\phi_{k,i',j} + w_{j+1}\tau_i$, where i' is the greatest value of i'', such that $(\tau_i - \tau_{i'}) \ge p_{j+1}$.

Combining the above two cases, we have following dynamic programming algorithm DP2:

Dynamic Programming Algorithm DP2 The Boundary Conditions:

$$\phi_{k,i,l=} \begin{cases} \min(e_{1}, w_{1}\tau_{j}), \text{ if } k \geq 1, p_{1} \in (\tau_{i-1}, \tau_{i}]; \\ w_{1}\tau_{j}, \text{ if } k = 0, \text{ and } p_{1} \in (\tau_{i-1}, \tau_{i}], \tau_{j} < \tau_{i}; \\ e_{1}, k \geq 1, p_{1} > \tau_{i}; \\ +\infty, others. \end{cases}$$

The Recursive Function:

$$\phi_{k,i,j} = \min(\phi_{k-1,i,j+1} + e_{j+1}, \phi_{k,i',j} + w_{j+1}\tau_i).$$

Now, observe that for finding an ε' -aligned schedule with the optimum objective function value, it is sufficient to assume that the completion time of the latest scheduled job is at most $(1+\varepsilon')^n \sum_{j=1}^n p_j$. The answer to our original problem is $\phi_{K,L,n}$, where K is the maximum number of jobs allowed to be rejected, L is the smallest integer such that $\tau_L \ge (1+\varepsilon')^n \sum_{j=1}^n p_j$. Thus, L is the smallest integer greater than or equal to $\log \sum_{j=1}^n p_j / \log(1+\varepsilon') + n$, whence $L = O(\frac{n}{\varepsilon} \log \sum_{j=1}^n p_j)$. So that the overall time for the dynamic program (FPTAS) is $O(nKL) = O((Kn^2/\varepsilon) \log \sum_{j=1}^n p_j)$. We also note that dividing each processing time and rejection penalty by the smallest processing time p < 1 increases the running time of the algorithm by at most a polynomial additive factor of $(Kn^2/\varepsilon) \log(1/p)$.

Theorem 3.5. Dynamic programming yields a FPTAS for problem $1 | rej, |R| \le K |\sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$, which runs in time $O((Kn^2 / \varepsilon) \log \sum_{j=1}^{n} p_j)$.

3.4. Discussions on Some Special Cases

Inspired by Lu *et al.* \cite{lu2021single}, we discuss some special cases about our first problem. For any instance $I = J_1, \dots, J_n$, let n_p, n_w and n_e be the numbers of distinct processing times, distinct weights and distinct rejection costs, respectively. In this subsection, we consider some special cases with $n_p = k$, $n_w = k$ or $n_e = k$, where k is a fixed constant. The corresponding problems are denoted $1 | n_p = k, |R| \le K | \sum w_j C_j + \sum_{J_j \in R} e_j$, $1 | n_w = k, |R| \le K | \sum w_j C_j + \sum_{J_j \in R} e_j$, $1 | n_e = k, |R| \le K | \sum w_j C_j + \sum e_j$. **Problem** $1 | n_p = k, |R| \le K | \sum w_j C_j + \sum e_j$

Suppose that a_1, a_2, \dots, a_k are k distinct processing times for instance $I = J_1, \dots, J_n$. Furthermore, we write

 $S_i = \{J_j : p_j = a_i\}$ and $|S_i| = m_i$. Next, we provide a polynomial-time algorithm for this problem.

First, similar to algorithm DP1, we obtain a new dynamic programming algorithm DP3 for problem $1 | rej, | R | \le K | \sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$.

To solve our problem, we set up a dynamic program for a harder problem: namely, to find the schedule that minimizes the objective function when the total processing time of the scheduled jobs and the total number of rejected jobs are given. We number the jobs in ascending order of p_j/w_j . Let $\phi_{k,t,j}$ denote the optimal value of the objective function when the jobs in consideration are $1, 2, \dots, j$, the total number of reject jobs is k, and the total processing time of the scheduled jobs is t.

Case 1. Job *j* is rejected. Then, the optimal value of the objective function is clearly $\phi_{k-1,t,j-1} + e_j$, since the total processing time of the scheduled jobs among $1, \dots, j-1$ must be *t*, the total number of reject jobs is k-1.

Case 2. Job *j* is scheduled. In this case, the total processing time of the scheduled jobs among $1, \dots, j-1$ must be $t - p_j$. Also, when job *j* is scheduled later all jobs in the optimal schedule for jobs $1, \dots, j-1$, the total weighted completion time is increased $w_j t$. Then, the optimal value of the objective function is $\phi_{k,t-p_i,j-1} + w_j t$.

Combining the above two cases, we have following dynamic programming algorithm DP3: **Dynamic Programming Algorithm DP3 The Boundary Conditions:**

$$\begin{cases} \phi_{k,t,1} = +\infty, \ k = 0, t \neq p_1; \\ \phi_{k,t,1} = e_1, \ k \neq 0, t \neq p_1; \\ \phi_{k,p_1,1} = w_1 p_1, \ k = 0; \\ \phi_{k,p_1,1} = \min\{w_1 p_1, e_1\}, \ k \neq 0. \end{cases}$$

The Recursive Function:

$$\phi_{k,t,j} = \min(\phi_{k-1,t,j-1} + e_j, \phi_{k,t-p_j,j-1} + w_j t).$$

Since the reject of the jobs can be at most *K*, the total processing time of the scheduled jobs can be at most $\sum_{j=1}^{n} p_j$, and the answer to our original problem is $\min\{\phi_{k,t,n}: 0 \le k \le K, 0 \le t \le \sum_{j=1}^{n} p_j\}$.

Theorem 3.6. Dynamic programming yields an $O(nK\sum_{j=1}^{n} p_j)$ time algorithm for solving problem $1 | rej, |R| \le K | \sum_{j \in A} w_j C_j + \sum_{j \in R} e_j$.

Specially, if all jobs have k distinct processing times a_1, \dots, a_k , then we have $t \in \{x_1a_1 + \dots + x_ka_k : 0 \le x_i \le m_i\}$. Thus, we have $O(\prod_{i=1}^k m_i) = O(n^k)$ choices for each t. As a result, we have the following corollary.

Corollary 3.7. Algorithm DP3 solves problem $1 | n_p = k$, $| R | \le K | \sum w_j C_j + \sum_{J_i \in R} e_j$ in $O(Kn^{k+1})$ time.

Problem 1 | $n_w = k$, $|R| \le K |\sum w_j C_j + \sum e_j$

Let b_1, \dots, b_k are the distinct weights. Furthermore, we set $S_x = \{J_j : w_j = b_x\}$ for each $x = 1, \dots, k$ and $|S_x| = m_x$. Specially, if all jobs have k distinct weights b_1, \dots, b_k , based on the DP1, we have $W \in \{y_1b_1 + \dots + y_kb_k : 0 \le y_x \le m_x\}$. Thus, we have $O(\prod_{x=1}^k m_x) = O(n^k)$ choices for each W. As a result, we have

the following corollary.

Corollary 3.8. Algorithm DP1 solves problem $1 | n_w = k$, $| R | \le K | \sum w_j C_j + \sum_{J \in R} e_j$ in $O(Kn^{k+1})$ time.

Problem 1 | $n_e = k$, | $R \models K \mid \sum w_j C_j + \sum e_j$

Note that in all algorithms DP1, DP2 and DP3, the current total processing time \$t\$ or total weight *W* of schedules jobs is always used to compute the w_jC_j value if J_j is accepted. Thus, it seems to be difficult to design an algorithm which does not include *t* or *W* as a parameter. Thus, we conjecture that problem $1 | n_p = k, |R| \le K | \sum w_jC_j + \sum_{J_j \in R} e_j$ is NP-hard even when $n_e = 1$. It might be a challenging problem to determine its exact computational complexity.

4. Scheduling under the Job Rejection Cost Constraint

In this section, we consider the problem $1 | rej, | R | \le K$, $\sum_{j \in R} e_j \le U | \sum_{j \in A} w_j C_j$. Our dynamic programming

algorithm and fully polynomial time approximation scheme (FPTAS) follow the ideas presented in [15].

We are given a list of jobs $\{J_j = (p_j, w_j, e_j): 1 \le j \le n\}$ and suppose that all the jobs have been indexed in non-decreasing order of p_j/w_j . The given threshold for job rejection constraint is U. Let f(j,k,P,A) be the minimum rejection cost of partial schedules for jobs J_1, J_2, \dots, J_j , whose total number of reject jobs is k, total processing time are P and objective function values are A. Let

$$p_{\max} = \max_{i \in I} p_i, w_{\max} = \max_{i \in I} w_i.$$

Case 1. Job *j* is rejected. Then, the optimal value of the objective function is clearly $f(j-1, k-1, P, A) + e_j$. In this case, when only the jobs J_1, \dots, J_{j-1} is considered, the total number of reject jobs is k-1 and the total rejection cost must plus e_j .

Case 2. Job *j* is scheduled. In this case, when the jobs J_1, \dots, J_{j-1} is considered, the number of the current accepted jobs is j-1 and the total rejection cost is not change. Furthermore, the total number of reject jobs is *k*. Thus, we have $f(j-1,k, P-p_i, A-w_iP)$.

Combining the above two cases, we have following dynamic programming algorithm DP3:

Dynamic Programming Algorithm DP3

The Boundary Conditions:

$$f(1,k,P,A) = \begin{cases} 0, \text{ if } P = p_1 \text{ and } A = w_1 p_1; \\ e_1, \text{ if } k \ge 1, P = 0 \text{ and } A = 0; \\ +\infty, \text{ others.} \end{cases}$$

The Recursive Function:

$$f(j,k,P,A) = \min\{f(j-1,k,P-p_i,A-w_iP), f(j-1,k-1,P,A) + e_i\}.$$

The optimal schedule can be obtained by finding the minimum *A* such that $f(n,k,P,A) \le U$ for some $0 \le k \le K$ and $0 \le P \le \sum_{j=1}^{n} p_j$, and derive the corresponding schedule by backtracking. The running time is $O(Kr^3 r^2 + rr^2)$

$O(Kn^3 p_{max}^2 w_{max})$.

Theorem 4.1. 1 | rej, $|R| \leq K$, $\sum_{j \in R} e_j \leq U | \sum_{j \in A} w_j C_j$ admits an FPTAS.

Proof: In order to get a $(1 + \varepsilon)$ -approximation in polynomial time, we need to use the trimming the state

space technique. The prove follows same idea as in [15]. Thus, we omit the detailed proof, the details are left to the interested readers.

5. Conclusions and Future Research

In this paper, we consider the single-machine scheduling with job rejection subject to the number of rejected jobs not exceeding a given threshold. We study two problems with rejection into our framework as follows: (1) the objective is to minimize total weighted completion time and the sum of rejection, (2) minimize total weighted completion time under the job rejection constraint. For the first scheduling problem, we provide a 2.618-approximation algorithm. In additions, we also discuss some special cases about our first problem. Furthermore, for the two problems, we provide a pseudo-polynomial time algorithm and a fully polynomial time approximation scheme (FPTAS), respectively.

In future research, an interesting direction is to consider our problems with release dates or submodular penalties. Moreover, it is also interesting to consider the online or semi-online versions of this problem. Finally, we also plan to extend this problem into parallel machine scheduling in the future.

Conflict of Interest

The authors declare that they have no known competing financial interests.

Author Contributions

Ruiqing Sun put forward the problems we want to study and the algorithm idea of the problems.

Mingyao Li mainly gives the relevant proof of the algorithm, and is responsible for drafting the paper.

Bin Deng is mainly responsible for reviewing and revising the paper, and is responsible for the final version of the paper.

References

- [1] Bartal, Y., *et al.* (2000). Multiprocessor scheduling with rejection. *SIAM Journal on Discrete Mathematics*, *13(1)*, 64-78.
- [2] Ou, J., Zhong, X., & Wang, G. (2015, March). An improved heuristic for parallel machine scheduling with rejection. *European Journal of Operational Research*, *241(3)*, 653-661.
- [3] Ou, J., & Zhong, X. (2017, Nov.). Bicriteria order acceptance and scheduling with consideration of fill rate. *European Journal of Operational Research*, *262(3)*, 904-907.
- [4] Sengupta, S. (2003). Algorithms and approximation schemes for minimum lateness/tardiness scheduling with rejection. *Proceedings of the Workshop on Algorithms and Data Structures* (pp. 79-90). Springer, Berlin, Heidelberg.
- [5] L., Zhang, Lu, L., & Yuan, J. (2009, Nov.). Single machine scheduling with release dates and rejection. *European Journal of Operational Research*, *198(3)*, 975-978.
- [6] C., He, *et al.* (2016, Jan.). Improved algorithms for single machine scheduling with release dates and rejections. *40R*, *14(1)*, 41-55.
- [7] Ou, J., Zhong, X., & Li, C.-L. (2016, Aug.). Faster algorithms for single machine scheduling with release dates and rejection. *Information Processing Letters*, *116(8)*, 503-507.
- [8] Zhang, L., & Lu, L. (2016, Jan.). Parallel-machine scheduling with release dates and rejection. *40R*, *14(2)*, 165-172.
- [9] Li, W., & Cui, Q. (2018, Mar.). Vector scheduling with rejection on a single machine. 40R, 16(1), 95-104.
- [10] Dai, B., & Li, W. (2020, Jan.). Vector scheduling with rejection on two machines. *International Journal of Computer Mathematics*, *97*(*12*), 2507-2515.

- [11] Liu, X., Li, W., & Zhu, Y. (2021, Aug.). Single machine vector scheduling with general penalties. *Mathematics*, *9*(*16*), 1965.
- [12] Liu, X., & Li, W. (2021, March). Approximation algorithms for the multiprocessor scheduling with submodular penalties. *Optimization Letters*, 1-16.
- [13] Liu, X., & Li, W. (2020, April). Combinatorial approximation algorithms for the submodular multicut problem in trees with submodular penalties. *Journal of Combinatorial Optimization*, 1-13.
- [14] Li, W., *et al.* (Nov. 2015). Penalty cost constrained identical parallel machine scheduling problem. *Theoretical Computer Science*, *607*, pp. 181-192.
- [15] Cao, Z., et al. (2006). On several scheduling problems with rejection or discretely compressible processing times. Proceedings of the International Conference on Theory and Applications of Models of Computation, (pp. 90-98). Springer, Berlin, Heidelberg.
- [16] Zhang, S.-X., Cao, Z., Zhang, Y. (2009, Sep.). Scheduling with rejection to minimize the total weighted completion time. *ISORA*, *9*, 111-114.
- [17] Zhang, Y., Ren, J., & Wang, C. (2009). Scheduling with rejection to minimize the Makespan. Proceedings of the International Conference on Combinatorial Optimization and Applications (pp. 411-420). Springer, Berlin, Heidelberg.
- [18] Liu, X., Yu, S., & Luo, W. (2020, Mar.). Approximation algorithms for single machine parallel-batch scheduling with release dates subject to the number of rejected jobs not exceeding a given threshold. *Operations Research Transactions, 24(1),* 131–139.
- [19] Zhang, X., *et al.* Approximation algorithms for precedence-constrained identical machine scheduling with rejection. *Journal of Combinatorial Optimization*, *35(1)*, 318-330, June 2018.
- [20] Zhang, L., Lu, L., & Yuan, J. (2010, Mar.). Single-machine scheduling under the job rejection constraint. *Theoretical Computer Science*, *411(16-18)*, 1877-1882.
- [21] Engels, D. W., *et al.* (Oct. 2003). Techniques for scheduling with rejection. *Journal of Algorithms, 49(1)*, 175-191.
- [22] Shabtay, D., Gaspar, N., & Kaspi, M. (2014, Feb.). A survey on offline scheduling with rejection. *Journal of Scheduling*, *16(1)*, 3-28.
- [23] Zhang, Y. (2020, June). A survey on job scheduling with rejection. *Operations Research Transactions*, 24(2), 111–130.
- [24] Graham, R. L., *et al.* (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, *5*, 287-326.
- [25] Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, *3(1-2)*, 59-66.
- [26] Woeginger, G. J. (2000, Feb.). When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)? *INFORMS Journal on Computing*, *12(1)*, 57-74.
- [27] Afrati, F., & Milis, I. (2006, Mar.). Designing PTASs for MIN-SUM scheduling problems. *Discrete Applied Mathematics*, *154*(*4*), 622-663.
- [28] Lu, L., Zhang, L., & Ou, J. (2021). Single machine scheduling with rejection to minimize the weighted makespan. *Proceedings of the International Conference on Algorithmic Applications in Management* (pp. 96-110).

Copyright © 2022 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (<u>CC BY 4.0</u>).



Ruiqing Sun is currently a master student in the School of Mathematics and Statistics, Yunnan University. His main research interest is combinatorial optimization.



Mingyao Li is currently a master student in the School of Mathematics and Statistics, Yunnan University. His main research interests are combinatorial optimization and Bayesian adaptive estimation.



Bin Deng is currently a master student in the School of Mathematics and Statistics of Yunnan University. His main research interest are combinatorial optimization and algorithmic game theory.